




GUI

Graficzny interfejs
użytkownika



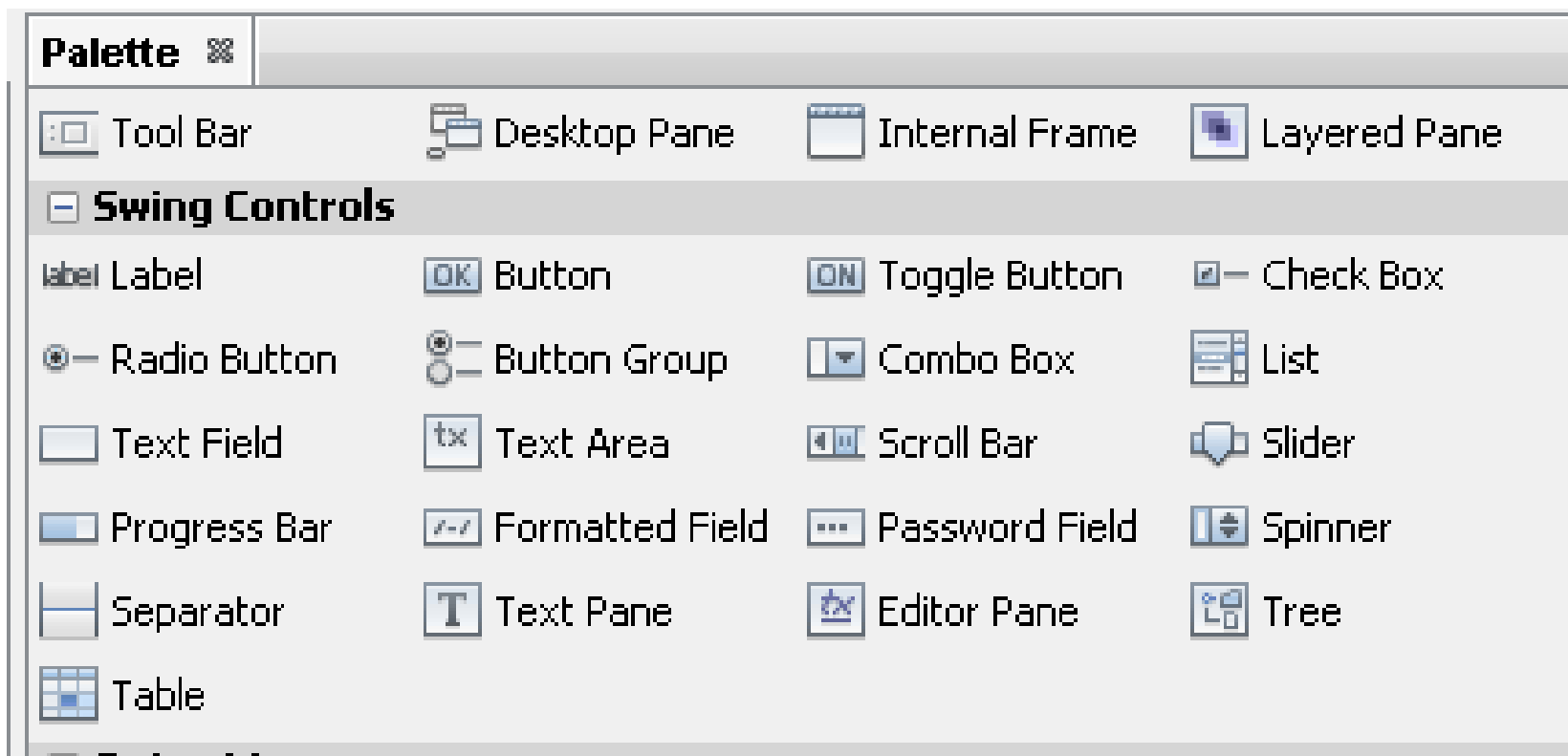
Podstawowe elementy GUI

The screenshot displays a Java IDE interface with the following components:

- File Explorer:** Shows open files: `Gui_TEST.java` and `mainWindows.java`.
- Source/Design/History:** A tabbed interface with the **Design** tab selected.
- Design Canvas:** A workspace containing a large empty rectangular area, a **Checkbox** component, and an **OK** button. Dotted lines indicate connections between the components.
- Palette:** A sidebar containing various GUI components categorized under **Swing Controls** and **Swing Menus**.
 - Swing Controls:** Label, Button, Toggle Button, Check Box, Radio Button, Button Group, Combo Box, List, Text Field, Text Area, Scroll Bar, Slider, Progress Bar, Formatted Field, Password Field, Spinner, Separator, Text Pane, Editor Pane, Tree, Table.
 - Swing Menus:** (Empty category)
- Properties Window:** Titled **jTextBox [JTextArea] - Properties**, with tabs for **Properties**, **Binding**, **Events**, and **Code**. The **Events** tab is active, showing a list of events and their values.

Event	Value
ancestorAdded	<none>
ancestorMoved	<none>
ancestorMoved	<none>
ancestorRemoved	<none>
ancestorResized	<none>

Paleta elementów



Podstawowe metody

- JLabel:
 - void setAlignment(Label.CENTER/LEFT/RIGHT) - wyrównanie
 - String getText() – pobiera aktualny tekst napisu
 - void setText(String text) – ustawia tekst napisu
 - void setDisplayedMnemonic(char aChar) - ustawia skrót klawiszowy
 - void setLabelFor(Component c) – ustawia do czego odnosi się label

Podstawowe metody

- `JRadioButton`
 - `boolean isSelected()` – czy wybrany
 - `void setSelected(boolean select)` – wybierz na sztywno
 - `add(ButtonGroup grupa)` – dodaj do grupy przycisków (aby można było wybrać jeden)
 - `setText()`, `getText()`, itp.
- `JComboBox`
 - `boolean isSelected()` – czy został wybrany
 - `void setSelected(boolean ustaw)` – zaznacz lub odznacz


Podstawowe metody

- `TextArea`

- `void append(String str)` – dopisz tekst do istniejącego
- `int getLineCount()` – ile linii tekstu jest w polu
- `void setFont(Font f)` – ustaw czcionkę
- `setLineWrap(boolean lineWrap)` – ustawia, czy linie będą będą łamane, czy nie
- `setText()`, `getText()`, itp..

Podstawowe metody

- JButton
 - `setEnabled(boolean akt)` – ustaw czy można go kliknąć, czy jest wyłączony
 - No ... jest 😊 Można go kliknąć i podpinać tzw. zdarzenia:

[JFrame] - Properties  **Palette**

Properties Binding **Events** Code

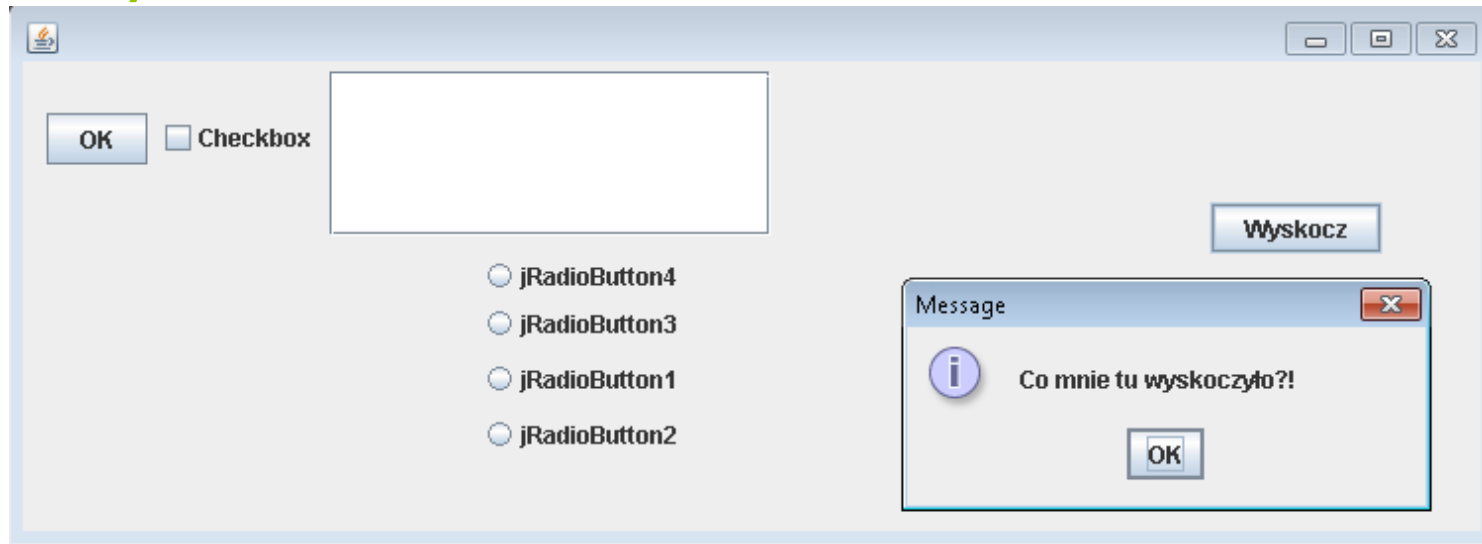
Events

ancestorMoved	<none>	▼	...
ancestorResized	<none>	▼	...
caretPositionChanged	<none>	▼	...
componentAdded	<none>	▼	...
componentHidden	<none>	▼	...
componentMoved	<none>	▼	...
componentRemoved	<none>	▼	...
componentResized	<none>	▼	...
componentShown	<none>	▼	...
focusGained	<none>	▼	...
focusLost	<none>	▼	...
hierarchyChanged	<none>	▼	...
inputMethodTextChanged	<none>	▼	...
keyPressed	<none>	▼	...
keyReleased	<none>	▼	...
keyTyped	<none>	▼	...
mouseClicked	<none>	▼	...
mouseDragged	<none>	▼	...
mouseEntered	<none>	▼	...
mouseExited	<none>	▼	...
mouseMoved	<none>	▼	...
mousePressed	<none>	▼	...
mouseReleased	<none>	▼	...
mouseWheelMoved	<none>	▼	...
propertyChange	<none>	▼	...
windowActivated	<none>	▼	...
windowClosed	<none>	▼	...
windowClosing	<none>	▼	...
windowDeactivated	<none>	▼	...

Podstawowe metody

- `CheckBox`
 - `void addItem(Object anObject)` – dodaje element do listy
 - `Object getItemAt(int idx)` – zwraca obiekt o indeksie `idx` (użycie to np. `String napis = combo.getItemAt(0).toString()`);
 - `int getItemCount()` – zlicza ilość opcji
 - `Object getSelectedItem()` – zwraca element, który jest wybrany
 - `void hidePopup()` – chowa wyskakujące okienko pola wyboru
 - `void removeItem(Object anObject)` – usuwa element (także inne wariacje: `removeItemAt(int idx)`, `removeAllItems()`)

Wyskakujące okienko, czyli konsola w GUI 😊



1. Zimportować `javax.swing.JOptionPane`;
2. Dodać zdarzenie `onMouseClicked()` do przycisku „Wyskocz”.
3. Wewnątrz napisać:
`JOptionPane.showMessageDialog(this, "Co mnie tu wyskoczyło?!");`

Ćwiczenia

1. Napisz program, który będzie pytał Cię o imię, a następnie po kliknięciu na przycisk – witał Cię z jego użyciem.
2. Napisz program kalkulator: użytkownik ma dwa pola, w którym wprowadza liczby oraz kilka pól jednokrotnego wyboru (radioButton) określających działanie.
3. Napisz program, w którym przycisk ucieka użytkownikowi spod kliknięcia
4. Zaimplementuj algorytm Luhna do weryfikacji numerów PESEL. Zadbaj o poprawny wygląd aplikacji. Dopisuj poprawne numery PESEL wprowadzane przez użytkownika w polu tekstowym do pliku tekstowego.